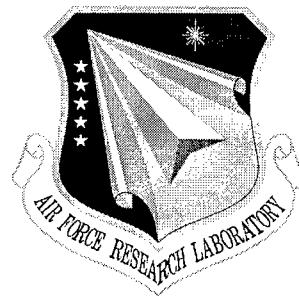


AFRL-IF-RS-TR-1998-132, Vol III (of three)
Final Technical Report
July 1998



CAMPAIGN PLANNING - User's Guide: System Design and User's Manual For SIPE-2 in IFD-4

SRI International

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. C671

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

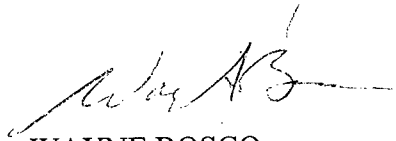
DTIC QUALITY INSPECTED 1

19981006 080

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.


AFRL-IF-RS-TR-1998-132, Vol III (of three) has been reviewed and is approved for publication.

APPROVED:



WAYNE BOSCO
Project Engineer

FOR THE DIRECTOR:



NORTHROP FOWLER, III, Technical Advisor
Information Technology Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFTB, 525 Brooks Road, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

USER'S GUIDE:
SYSTEM DESIGN AND USER'S MANUAL FOR SIPE-2 IN IFD-4

Marie Bienkowski

Contractor: SRI International
Contract Number: F30602-95-C-0175
Effective Date of Contract: June 1995
Contract Expiration Date: September 1997
Program Code Number: 7E20
Short Title of Work: SOCAP-ACPT Technology Integration
Period of Work Covered: Jun 95 - Sep 97

Principal Investigator: Marie Bienkowski
Phone: (650) 859-5485
AFRL Project Engineer: Wayne Bosco
Phone: (315) 330-3578

Approved for public release; distribution unlimited.

This research was supported by the Defense Advanced Research
Projects Agency of the Department of Defense and was monitored by
Wayne A. Bosco, AFRL/ITB, 525 Brooks Road, Rome, NY
13441-4505.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE July 1998		3. REPORT TYPE AND DATES COVERED Final Jun 95 - Sep 97
4. TITLE AND SUBTITLE CAMPAIGN PLANNING - User's Guide: System Design and User's Manual for SIPE-2 in IFD-4			5. FUNDING NUMBERS C - F30602-95-C-0175 PE - 62301E, 63278F PR - C671 TA - 00 WU - 01	
6. AUTHOR(S) Marie Bienkowski				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) SRI International 333 Ravenswood Avenue Menlo Park CA 94025-3493			8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency Air Force Research Laboratory/IFTB 3701 North Fairfax Drive 525 Brooks Road Arlington VA 22203-1714 Rome NY 13441-4505			10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-1998-132, Volume III (of three)	
11. SUPPLEMENTARY NOTES Air Force Research Laboratory Project Engineer: Wayne A. Bosco/IFTB/(315) 330-3578				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Volume III is a system design report and user's manual for SIPE-2 in Integrated Feasibility Demonstration -4 (IFD-4). This report discusses the operational model used for feasibility estimation in IFD-4. Described are the systems modules and the SIPE -2 user interface. This report further discusses the integration design and implementation of SIPE-2, Air Campaign Planning Tool (ACPT) and Conventional Target Effective Model (CTEM). Details are provided on the installation and operation of the IFD-4 software.				
14. SUBJECT TERMS SIPE, Integrated Feasibility Demonstration -4 (IFD-4), Conventional Target Effective Model (CTEM), Air Campaign Planning Tool (ACPT)			15. NUMBER OF PAGES 40	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

CONTENTS

1	OVERVIEW	1
2	OPERATIONAL MODEL.....	2
3	SYSTEM MODULES.....	5
4	SIPE-2 USER INTERFACE.....	7
4.1	SETTING SITUATION OPERATIONAL PARAMETERS.....	9
4.2	MANIPULATING THE RULES OF ENGAGEMENT.....	12
4.3	CHANGING THE PLANNING PREFERENCES.....	12
4.4	INVOKING SIPE-2 TO PLAN OR ANALYZE FEASIBILITY.....	12
4.5	TEXT VIEWS OF THE PLAN.....	12
4.6	PLAN COMPARISON USER INTERFACE.....	14
5	SIPE-2:ifd-4 DESIGN AND IMPLEMENTATION.....	16
5.1	LOGICAL PATHNAME TRANSLATIONS.....	16
5.2	SIPE-2:IFD-4.....	17
6	SIPE-2 AND ACPT.....	18
7	SIPE-2 AND CTEM.....	19
8	INSTALLING AND OPERATING THE IFD-4 SOFTWARE.....	20
9	REFERENCES.....	24

FIGURES

1	IFD-4 System Modules.....	3
2	IFD-4 Feasibility Estimation Phases.....	6
3	Normal SIPE-2 Interface Window.....	8
4	SIPE:ifd-4 Control Panel Window.....	10
5	SIPE:ifd-4 Resource Table Window.....	11
6	SIPE:ifd-4 Text Plan View Window.....	13
7	Plan Resource Comparison.....	15
8	PVT Display Showing Three Levels of the Plan.....	22
9	PVT Display Showing Support Missions Generated by SIPE-2:ifd4.....	23

TABLES

1	Subdirectories of the sipe-ifd Directory.....	16
2	Source Code Files for SIPE-2:ifd-4.....	17
3	Source Code Files for SIPE-2:ifd-4.....	18

1 OVERVIEW

Most research in artificial intelligence (AI) planning has focused on the problems involved with rapidly developing quality plans. The tools resulting from this research have been applied in numerous domains, and, in most large-scale military domains, these applications have highlighted the need for software to complement the "reasoning about actions" capabilities of AI planners. AI planners are useful for handling details in reasoning about networks of cause and effect; dependencies among actions within the plan; constraints on resources available for actions (including actors); and plan repair. On the other hand, they generally lack intuitive user interfaces and strong scheduling and temporal reasoning capabilities. In military applications we have found that these capabilities can be provided by legacy systems (i.e., systems already in operation), leading to the need for the integration of AI planning applications with these legacy systems, so that the practical utility of AI planning in large-scale military domains can be demonstrated.

SIPE-2* is a domain independent, state-of-the-practice AI planning system that has been applied in a number of military domains. Under the Advanced Research Projects Agency (now DARPA)/Rome Laboratory Planning Initiative (ARPI) program, it was an integral part of the fourth Integrated Feasibility Demonstration (IFD-4), which showed the feasibility of AI planning in support of plan development and minor editing, plan refinement, and feasibility estimation for air campaign planning for the United States Air Force (USAF). Its most powerful contribution was the ability to create and maintain a complex plan representation. In the case of IFD-4, this representation combined parts of several other reasoning systems.

AI planning adds valuable functionality to end-to-end air campaign planning, especially in feasibility estimation, which is the process of determining if a plan, articulated to any level of detail, can be accomplished, given practical operating conditions such as available resources (including logistics, personnel, and intelligence assets). Feasibility estimation acts as a reality check while the user plans either the strategic or the tactical parts of a mission. In either situation, AI planners can provide assistance by filling out incomplete parts of the plan, generating detailed activities that define the more abstract actions and goals of the plan, and extracting plan elements for analysis by an external module such as a logistics scheduler. In addition to serving as a prelude for analysis, generating the detailed plan itself serves as a check of the plan's feasibility: if, for example, no detailed plan can be generated, then assumptions were made in higher-level planning that could not be satisfied in more detailed planning.

SIPE-2 was critical to the feasibility estimation conducted for IFD-4. During IFD-4, SIPE-2 was integrated with several other AI or USAF legacy systems, and provided a central point for the maintenance of a hybrid plan representation. This hybrid plan combined elements from these other systems. SIPE-2 created detailed plans derived from the user's high-level strategic goals, extracted information from the resulting plans for feasibility estimation, and allowed the user to create a selected set of plan modifications. Bienkowski [1997] gives more details on SIPE-2's role in IFD-4. Wilkins [1997] gives specific details on using SIPE-2. SIPE-2 also uses the Advisable Planner module, which is described by Myers [1996]. Bienkowski, Lee, and Wolverton [1997] provide details on the SIPE-2 knowledge base for IFD-4. This user's guide covers only the design

*SIPE-2: System for Integrated Planning and Scheduling. SIPE-2 is a trademark of SRI International. All product or company names mentioned in this document are the trademarks of their respective holders.

and operation of SIPE-2 as it is applied to air campaign feasibility estimation (which we henceforth will refer to as SIPE-2:ifd-4), extensions made to SIPE-2 for IFD-4 (including integration code), and the SIPE-2:ifd-4 user interface.

In Section 2 of this user guide, we discuss the operational model for feasibility estimation in IFD-4. In Section 3, we describe the IFD-4 system modules; Section 4 describes the SIPE-2 user interface. In Section 5 we describe the integration design and implementation. Sections 6 and 7 describe the integration of SIPE-2 and APCT, and SIPE-2 and CTEM, respectively. Section 8 discusses the installation and operation of the IFD-4 software.

2 OPERATIONAL MODEL

IFD-4 was a proof-of-concept demonstration of advanced technology application in support of plan feasibility estimation, and was developed under the guidance of the USAF's Checkmate Office. Figure 1 shows a diagram of the modules that make up IFD-4.

During the development of an air campaign plan, SIPE-2:ifd-4 provides the campaign planner with a rough estimate of the feasibility of a partial or complete air campaign plan. Campaign planning in IFD-4 began with the use of the Air Campaign Planning Tool (ACPT^{*}), a plan authoring tool used to support the top-down, strategy-to-task style of planning performed by Checkmate. ACPT allows users to input plan objectives, and then successively decompose them to different planning levels such as national, political, and military. In IFD-4,[†] a plan for a fictitious scenario was created, and most of the plan was articulated by the ACPT user down to the levels of air objectives, air tasks, and targets. However, the Air Superiority air objective was not planned in more detail because SIPE-2:ifd-4 completes that part of the plan.

After having completed this much of the plan, the user elected to conduct a feasibility estimation by bringing up the window for SIPE-2:ifd-4. In our design, the newly created plan would be downloaded into SIPE-2:ifd-4 from ACPT,[‡] at which point the user would see the plan, in a text view, or could examine or alter the resources available for the campaign on each day of the operation.^{**} SIPE-2:ifd-4 continued the development of the campaign plan by expanding the Air Superiority objective so that the entire plan terminated in targets. During this plan expansion, SIPE-2:ifd-4 relied on the Tachyon temporal reasoner^{††} to propagate temporal constraints.

Once a full target set was established, SIPE-2:ifd-4 interfaced with the conventional targeting evaluation model (CTEM)^{‡‡} to assign strike aircraft and munitions to targets, and to select a specific time for each strike to occur. Results from CTEM were incorporated into the

^{*} ACPT was developed by ISX Corporation (ISX) with the active participation of the USAF Checkmate Planning Office.

[†] IFD-4 also demonstrated the use of a plan critiquing method called INSPECT, developed by ISI. For simplicity, we will ignore this part of IFD-4, since it was separate from the feasibility estimation part of IFD-4.

[‡] In the actual IFD-4 demonstration, however, we had the plan preloaded in order to shorten the demonstration time.

^{**} The ability to alter resources then compute a feasibility estimation results in a what-ifing capability.

^{††} The Tachyon temporal reasoner was developed by GE Corporate Research and Development.

^{‡‡} CTEM is a legacy system used in planning by Checkmate, and its use in IFD-4 increased the credibility of the results with the users.

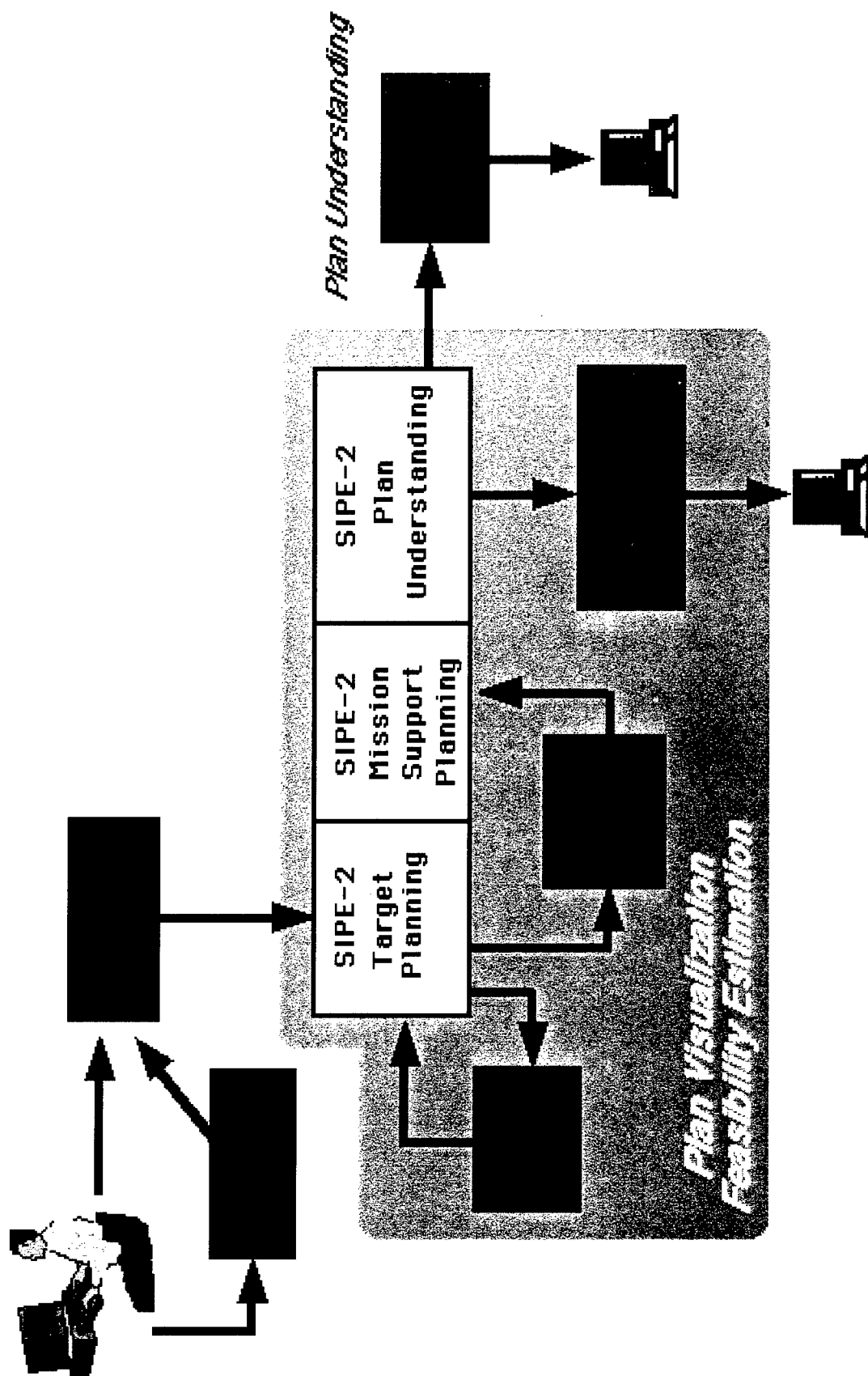


Figure 1. IFD-4 System Modules

SIPE-2:ifd-4 plan, resulting in a plan representation that used elements from ACPT, Tachyon, SIPE-2, and CTEM. SIPE-2's ability to handle these diverse representations shows the power and flexibility available with its use.

When the CTEM results were incorporated into SIPE-2:ifd-4, more detailed planning was required to ensure feasibility. SIPE-2:ifd-4 completes the plan expansion using a knowledge base of mission support operations to add the necessary escort, refueling, suppression of enemy air defense (SEAD), and intelligence/surveillance/reconnaissance (ISR). If SIPE-2:ifd-4 is successful in planning to this level of detail (i.e., if it had enough resources to assign to the actions it planned), the plan was considered feasible.

However, the user cannot see the results of this planning through the interfaces designed for SIPE-2:ifd-4: during plan expansion, no output is shown. After expansion is completed, though, the user may view a text-based, hierarchical presentation of the plan through a user interface. A limited number of plan modifications are also permitted.* For IFD-4, a user could add a new objective to the plan through this interface, and then rerun the plan's expansion feasibility estimation. At this point, the user could also alter resources (removing F-117s for the duration of the operation was shown for IFD-4), and then rerun the estimation.

Therefore, in order to check feasibility, the user needs to see a more detailed view of the plan. IFD-4 used a timeline-based and color-coded view of the expanded plan from SIPE-2:ifd-4 called the Plan Visualization Tool (PVT). The user selected the PVT window, which read from a file written by SIPE-2:ifd-4. That file contained the details of all of the actions at every plan level, plus annotations that provide (1) an English sentence describing each action, and (2) a color indicating the feasibility of that action.

The next, less important, part of IFD-4 was the demonstration of limited plan modification capabilities. SIPE-2 has the ability to add or delete goals from a plan during or after planning: operationally, this means that an interface must be provided such that the user can indicate which goals to add, and which to delete. For IFD-4, SIPE-2:ifd-4 permitted the deletion of goals (i.e., to indicate a change in the situation) based on the text-based display, after which the plan could be re-expanded and rechecked.

A final capability was added to the system after the IFD-4 demonstration, as follows. When the user changes a plan either by adding or deleting a goal, or by changing the available resources, SIPE-2:ifd-4 can recompute the expanded plan and feasibility. Naturally, the user wants to know what the differences are between the old and new plans, so a resource-based plan comparison feature was added with user interface windows in order to display the differences.

IFD-4 demonstrated the practical application of some of SIPE-2's extensive capabilities: completing a partially specified plan; responding to changes in resource availability; and interacting with a model that provided scheduling information.

*The set is strictly limited to a few operations.

3 SYSTEM MODULES

In this section, we discuss each of the separate system modules in the IFD-4 software developed by SRI, so that we may show the extent of the integration effort. We have divided the operation of this software into distinct phases, as shown in Figure 2. Each phase accomplishes a significant integration or plan generation operation.

In the first phase, SIPE-2 downloads a plan from ACPT by connecting (via Lisp code provided by ISX) to a module called the Node Object Manager (NOM). The NOM provides an interface to the ACPT object database (in which the plan, as created by the ACPT user, is stored). The ACPT plan is translated into a hierarchical list of structures, implemented in Lisp using the Common Lisp Object Structure (CLOS), that contains ACPT plan data down to the level of targets. Our design allows SIPE-2 to traverse this list to find those goal nodes that are unexpanded: in practice, we simply looked for the Air Superiority goal, since this was the one that SIPE-2 solved for IFD-4. That goal was translated into a SIPE-2 goal by SIPE-2's ACT plan representation (which permits ASCII representations of planning problems). SIPE-2 retains a copy of the entire CLOS plan, and connects to it by way of links that point from SIPE-2 structures to corresponding CLOS structures.

In the second phase, the Air Superiority portion of the plan was planned. In this phase, SIPE-2:ifd-4 uses a knowledge base developed for air campaign planning. SIPE-2 stops planning when it reaches the target level (it is programmed to think that it has, at this point, completed a plan). During this phase (at the end of every planning level) SIPE-2 calls Tachyon. SIPE-2 writes a file of temporal constraints; invokes Tachyon, which reads the file; and writes out a file of updated temporal constraints.

In Phase 3, SIPE-2 interacts with the legacy system, CTEM, in order to schedule missions. In our design, we anticipated that CTEM would be called during the demonstration. SIPE-2 would be able to extract relevant information from the targets in its plan and write a file for CTEM, at which point CTEM would use situation-specific data files (not written by SIPE-2) along with this information provided by SIPE-2, and would calculate airframes, munitions, and schedule. In practice, CTEM ran so slowly (one-half to one hour) that running it during the demonstration was not practical. Thus, we precomputed three different CTEM runs: one with the full resource set, one without F-117s (to show a resource shortfall), and one with a goal added to attack weapons of mass destruction (WMDs). During the demonstration, SIPE-2 reads these precomputed CTEM output files, and selects from among them based on where execution is in the demonstration script. SIPE-2 incorporates the results of the CTEM run into its target-level plan: at this point, the plan appears to SIPE-2's internal machinery as an incomplete plan, terminated in target packages (combined from ACPT and CTEM), with goals for generating the support missions for the packages.

After Phase 3 and the incorporation of CTEM output, SIPE:ifd-4 continues the development of the plan. It adds support missions, using a knowledge base of such operations developed for IFD-4. The feasibility estimation is then complete, and in Phase 4, SIPE:ifd-4 creates output from the hybrid plan representation to display in PVT.

In the final phase, the user may (1) view a text display of the plan via SIPE:ifd-4, (2) see the resources used in one plan, (3) see a comparison of the resources used in two plans, and (4) delete goals from a plan. Air tasks can also be written for ACPT to read, thus completing the cycle of planning from ACPT to SIPE:ifd-4. The specific user interfaces are described in the following sections.

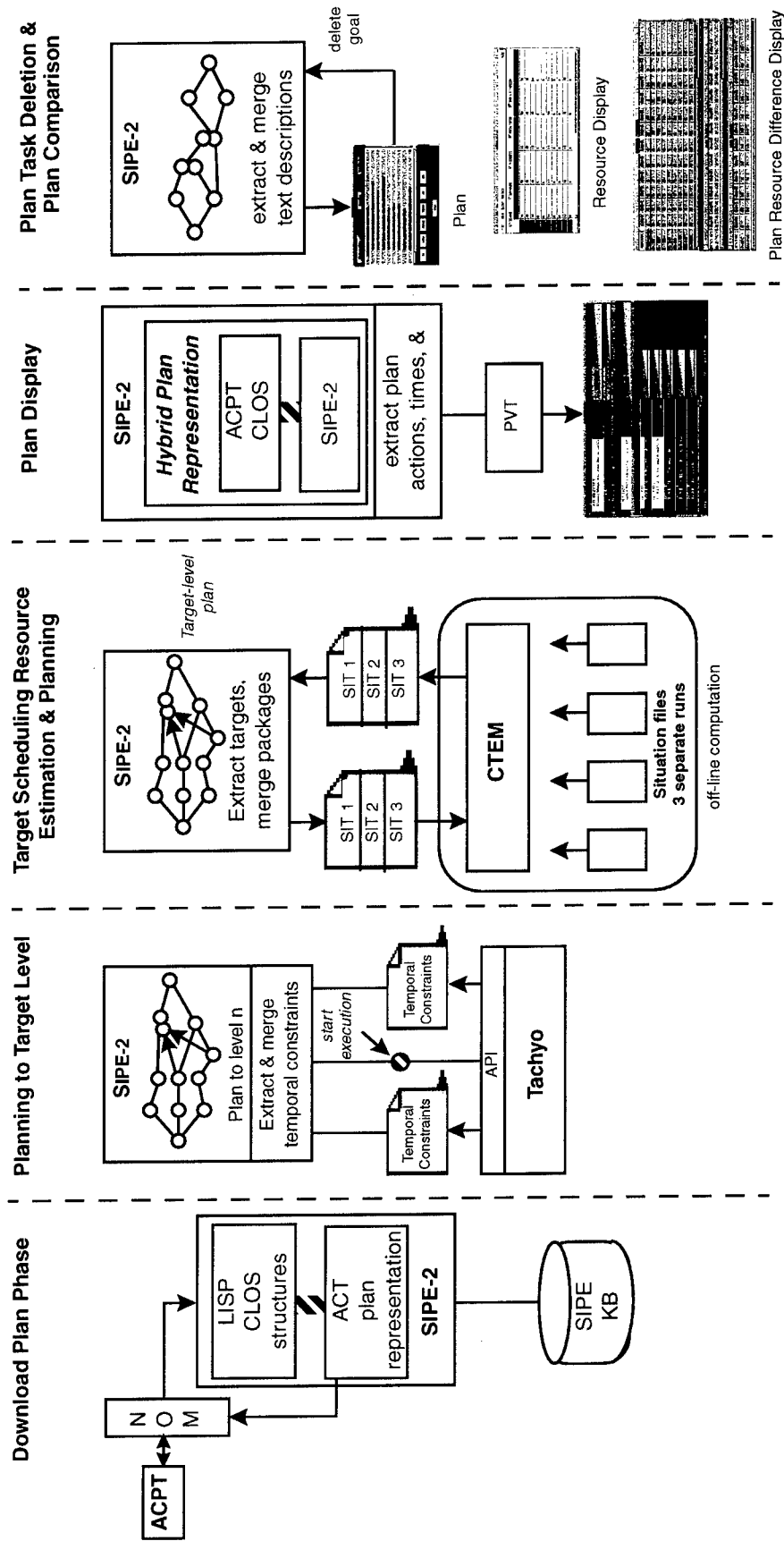


Figure 2. IFD-4 Feasibility Estimation Phases

System Software. The remainder of this guide describes system software and operation for SIPE:ifd-4. The operation of SIPE:ifd-4 requires the installation of Allegro Common Lisp Version 4.3 and SIPE-2 version 4.13. (It may work with later SIPE-2 releases, but the software [as delivered] was tested with version 4.13, V 1.11 of ACT, and V 1.23 of Grasper.) The software release is provided as one tar file, out of which can be extracted 5 directories and a README* file. These directories are (1) `sipe-ifd4`, which contains the Lisp code for SIPE:ifd-4 that is to be loaded on top of SIPE-2, (2) `userinterface`, which contains the source, binaries, and data files for the Motif-based user interface to SIPE:ifd-4, (3) `acptinterface`, which contains the Lisp files that (when loaded on top of SIPE:ifd-4) permit an ACPT-created plan to be downloaded into the SIPE:ifd-4 Lisp image, (4) `cteminterface`, which contains the data files for interfacing with CTEM, and (5) `tachyon-files`, which contains a few initialization files for operating the PVT part of Tachyon.

In the following sections, we describe the SIPE-2 user interface, the SIPE:ifd-4 software, and the ACPT and CTEM interfaces. Details on operating SIPE-2, including its interface to Tachyon, can be found in Wilkins [1997].

4 SIPE-2 USER INTERFACE

A number of IFD-4-specific user interfaces were added to SIPE-2 to support the display of results in a form acceptable to Checkmate. Figure 3 shows the typical SIPE-2 interface (details on the operation of SIPE-2 through this interface can be found in Wilkins [1997]). In order to make SIPE-2 compatible with the look and feel of the ACPT user interface, we used Motif as the interface standard, creating windows with the UIM/X tool.[†] UIM/X was especially helpful in creating complex objects such as tables. The interface is implemented by means of two separate UNIX processes,[‡] which can be started either from the command line or from a call to the command line from Lisp (the SIPE-2 implementation language). SIPE-2 and the UI are connected by means of an Unreliable Data Protocol (UDP) function: the user interface commands read from and write to files in order to pass data to and from SIPE-2.

In the software distribution, the user interface source and command executables are in a top level directory called `userinterface`. The source code is in a subdirectory** called `src`, and executables and data files for the interface are in a directory called `uiifd4`. We describe each command and its associated windows and input/output (I/O) files below.

*In this document we will use the following font conventions. Courier font is used to indicate file or directory names in the software distribution, commands to type for software operation, or buttons to press on the user interface. Brackets are used to indicate arguments to commands. A vertical bar between commands (e.g., `File|Quit`) indicates a sequence of selections from a menu.

[†]UIM/X is a product of Bluestone Software, Inc.

[‡]These two separate programs will appear to the user as one. Two programs are required in order to make use of the table feature of UIM/X.

**To enable the use of this source code, a license for, and installation of, UIM/X is required.

-PLAN-1

Drawings

APPLICATION

PROFILE

DOMAIN

PLAN

DRAWINGS

NODE

ADVICE

DRAW:

operator graph

operator

problem

plan

plan-to-node

world

objects

SELECT

DESTROY

DESTROY-multiple

NEW VIEW

RENAME

REDRAW

REPARSE

BACKUP

REVERT

RESCALE

HARDCOPY

GRAPH:

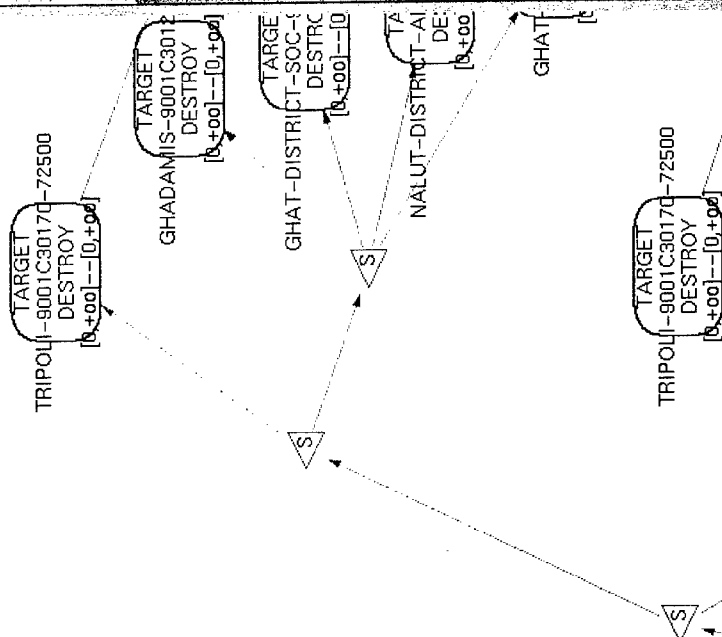
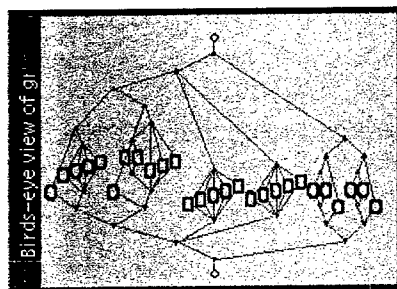
select

create

destroy

input

output



Command: :Application Mode
Command: :Drawings Mode
Command: :Plan136
Command: :Plan136
Creating grasper space AS-7-PLAN-1.
Command: □

AS-7-PLAN-1 in sipe.graph, Sipe-2 -- Unpublished (C) 1986-1995, SRI International

M: Manipulate the birds-eye view; R: Send the mouse pointer to the command menu.

Figure 3. Normal SIPE-2 Interface Window

`uisafe`: This command produces a main display (or control panel) window, shown in Figure 4, that controls the execution of SIPE-2:ifd-4. It is invoked as `uisafe` or `uisafe [hostname]` where `hostname` is the name of the machine on which the SIPE-2 process is running. If `hostname` is not specified, the UDP connection will be made on the local host.

Items to Note. When exiting the application via the `File | Close` button in the control panel window, the windows will disappear but the process may not completely exit. Therefore, simply type in `ctrl-C` to terminate the program.

The control panel window offers five operations. These are described below, roughly in the order in which they would be selected in an operational mode.

4.1 SETTING SITUATION OPERATIONAL PARAMETERS

From the control panel, select `Situation | Blue Forces`. This is the only option currently available, and only the `Aircraft` option is available for `Blue Forces`. The remainder of the `Situation` selections are intended to suggest the type of situation changes allowed by a fully implemented planning system. These are `Red Forces`, `Red Intentions`, and `Weather`.

Selecting `Situation | Blue Forces | Aircraft` produces the interface window shown in Figure 5.* This window is the resource table, and shows the resources (in this instance, aircraft) used over time.

- When `uitable` brings up the resources table window to populate the table cells, it reads from a data file called `acschedule.dat`. This file is used for communication between the user interface and SIPE-2:ifd-4.
- An entire row can be zeroed out, as well as a single column or the entire table, as follows:
 - 1. Click on the desired row or column label (e.g., F-117A or Day 1) to highlight that entire row/column.
 - 2. From the menu, select `Table`. The options `clear row`, `clear column`, or `clear entire table` (which will replace the indicated cells with 0s) will be presented.
- The table can be reloaded with the original data set (since the last invocation of the `File | Save` option) by selecting the `Table | Reload Table` option.
- Changes may be made to the table, but in order for SIPE-2:ifd-4 to collect information properly, the changes must be via this window. In order to save changes, select `File | Save`, which will overwrite the current contents of the `acschedule.dat` file.
- To exit this window, select `File | Quit`.

Items to Note. The system sometimes takes a while to create this window, but has no “waiting” cursor to indicate that you should wait, at this time. Clicking on the button again causes multiple `uitable` resource table windows to appear. To delete these multiple windows, close them using the `Quit` command.

*Because we have not been able to properly integrate the XRT/Table widget package into UIM/X, this window is actually a separate application, called `uitable`, that is invoked from the main `uisafe` program via a system call.

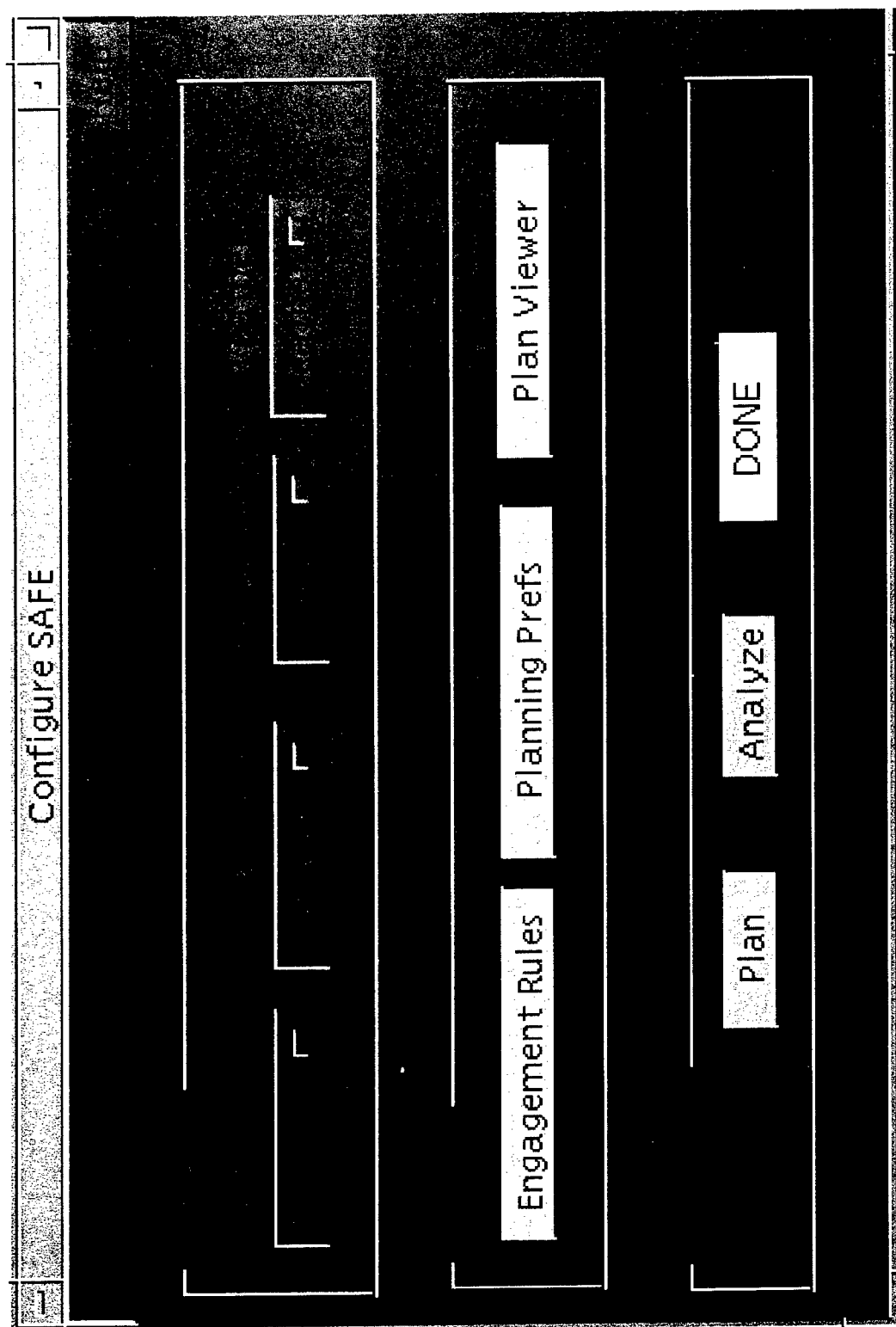


Figure 4. SIPE:ifd-4 Control Panel Window

File Table Base Sorties							Help
<input type="checkbox"/> Deployed	<input checked="" type="checkbox"/> EnRoute	<input checked="" type="checkbox"/> Allocated	<input checked="" type="checkbox"/> Requested	<input checked="" type="checkbox"/> World Inventory			
						Day 5	
18	18	18	18	18		18	
48	48	48	48	48		48	
18	18	18	18	18		18	
6	6	6	6	6		6	
6	6	6	6	6		6	
14	14	14	14	14		14	
0	0	0	0	0		0	
0	0	0	0	0		0	
12	12	12	12	12		12	
0	0	0	0	0		0	

Figure 5. SIPE:ifd-4 Resource Table Window

4.2 MANIPULATING THE RULES OF ENGAGEMENT

In order to suggest modifications that could be made to the rules of engagement in a fully implemented feasibility estimator, the `Engagement Rules` button on the control panel window brings up another window showing a form with rules and fields for entering integer values. Although the values are not used by SIPE-2:ifd-4, values for each field may be entered. Clicking the `OK` button closes this window.

4.3 CHANGING THE PLANNING PREFERENCES

The control panel selection `Planning Preferences` brings up a window that shows the possible preferences (i.e., advice) that can be passed on the feasibility estimator in a fully implemented system. These preferences deal with the scope of guidance, the amount of acceptable collateral damage risk, etc. These values are not currently used by SIPE-2:ifd4.

4.4 INVOKING SIPE-2 TO PLAN OR ANALYZE FEASIBILITY

From the control panel window, two `Execution` buttons are available that will invoke the SIPE-2:ifd-4 in two different ways. The first button generates a plan, starting from an Air Superiority objective and working down to the target level, without invoking CTEM to add packages to the target. This is called the pre-CTEM plan, and is created with the `Plan` button. Clicking on the `Analyze` button, however, will start planning from the current SIPE-2:ifd-4 plan (which is assumed to be a pre-CTEM plan), call CTEM, and then complete the plan, which is then called the post-CTEM plan.

When either button is selected, relevant user interface data is written to the file `uioutput`. This file is read by SIPE-2:ifd-4 when it receives the UDP message from `uisafe`. While SIPE-2:ifd-4 is running, the lower right-hand button in the control panel window is labelled `Busy`. Once SIPE-2:ifd-4 has completed processing, it sends a "done" message back to the user interface. The user interface then changes the label on the lower right-hand button in the control panel to `Done`.

4.5 TEXT VIEWS OF THE PLAN

From the SIPE-2:ifd-4 interface, a user can also elect to see a text view of the plan. The control panel has a `Plan Viewer` button, which, when selected, creates a window as shown in Figure 6. The plan is read by the user interface code from a file called `planFile.txt`, which contains the text for the objectives in a hierarchical form.

Three buttons in this window are functional. The `Display` menu has four options, the top level option indicating the highest level of objectives. When the window first appears, selecting any one of these options brings text into the display portion of the window. The `Add Objectives` button brings up a dialog box that allows the user to type in text that represents an objective.* When the user confirms the input by clicking the `OK` button, the text appears as a new objective at the end of

*This text is not parsed or used by SIPE-2:ifd-4. It is simply recorded, then passed to ACPT.

the current list displayed. Finally, the `Save Plan` command writes out the tasks and objectives to an ACPT-formatted file called `savePlan.txt`. This file can be read by ACPT in order to record the new objectives.

Items to Note. The display menu does not return to the default Display Objective (the first menu item) when reinvoking the Plan Viewer window; instead, it remains at the most recently selected option. This can be somewhat misleading, since the first level (Objectives) is always displayed when the window comes up. The menu selection can be changed simply by reselecting the desired option.

4.6 PLAN COMPARISON USER INTERFACE

Three other user interface commands together make up the display* windows for the resource-based plan comparison tool. They each take a file name as an argument and use the name of the file as the title of the table (see Figure 7). Table values are displayed in the rows in the file. The first process, `commonResources`, creates a Motif table that shows common resources; the second, `Plan-A-Not-B`, shows resources that are in the second plan but not the first; and the third, `Plan-B-Not-A`, shows resources that are in the first plan but not the second. An experienced planner can readily read and absorb from these details in order to succinctly characterize the plan differences.

*The use of the term “display” is intended to indicate that the window does not permit interaction, e.g., editing values with in a table or clicking buttons to initiate an operation.

Common Courses						
	day-1	day-2	day-3	day-4	day-5	day-6
	0	0	0	0	0	6
	0	0	0	0	0	29
	0	0	0	0	0	117
	0	0	0	0	0	0
	0	2	2	2	2	54
	0	0	0	0	0	6
	0	0	0	0	0	58
	0	0	0	0	0	6
	0	0	0	0	0	25
	0	0	0	0	0	0

Plan-A, Vol. 3						
	day-1	day-2	day-3	day-4	day-5	day-6
	0	0	0	10	10	10

Plan-B, Vol. 3						
	day-1	day-2	day-3	day-4	day-5	day-6
						5

Figure 7. Plan Resource Comparison

5 SIPE-2:IFD-4 DESIGN AND IMPLEMENTATION

SIPE-2:ifd-4 played a central role in the integration effort that constituted IFD-4. The integration process is described in detail by Bienkowski [1997]. In this section, we describe the major functions that were added to SIPE-2 in order to accomplish this integration and produce IFD-4, and also describe how these functions operate and provide administrative details for their use.

In the SIPE-2:ifd-4 distribution, the files required for building SIPE-2:ifd-4 are in a directory called `sipe-ifd4`. Table 1 shows the subdirectories of `sipe-ifd4`.

Table 1. Subdirectories of the `sipe-ifd` Directory

FILE	DESCRIPTION
README	An ASCII text file describing the directory contents
bin-sipe	SIPE-2 images with IFD-4 code and an ACPT-plan already loaded
data	Files created by CTEM and input to SIPE-2:ifd-4
doc	Scripts and incidental papers
kb	The SIPE-2 IFD-4 knowledge base
lib	C libraries loaded for Motif interface functions
lisp	Source code for IFD-4 (loaded on top of a SIPE-2 Lisp image)

The source code in `lisp` adds functionality to SIPE-2 by using existing SIPE-2 primitives. When the source code in `lisp` is loaded (with perhaps an ACPT plan copied in—see Section 6) a Lisp image may be saved for rapid testing and demonstration. These executable images are stored in `bin-sipe`, and are called either `ifd4-final` or `sipe-ifd4.5` in the distribution of SIPE-2:ifd-4.

Three other directories are required for SIPE-2:ifd-4 operation: one contains the user interface processes and was described in the previous section; the file `lib/udplisp.so` is loaded into SIPE-2:ifd-4 to provide a UDP-based connection between Lisp and Motif; and the `data` directory contains files that are produced by CTEM for SIPE-2:ifd-4 (and vice versa).

5.1 LOGICAL PATHNAME TRANSLATIONS

Careful attention must be paid to pathnames in order to operate the demonstration software correctly. As we have already noted, the demonstration relies on installations of both Allegro Common Lisp and SIPE-2, which have their own pathname dependencies. Pathnames relative to the new installation location of SIPE-2:ifd-4 should be set in the file `lisp/translations.lisp` when the software is installed. The following list shows the pathname translations, as they appear in the file `lisp/translations.lisp` in the software distribution. It is this file into which any edits may need to be made.

- (`"acpt;*. *.*"`
 `"/aaftp/mcclin/7112-delivery/acptinterface/JPT/JPTSystem/jpt/src/UI`
 `/cAgent/"`)
- (`"useri;*. *.*"` `"/aaftp/mcclin/7112-delivery/userinterface/uiifd4/"`)

- (`ctem-bin;*.***` `/aaitp/fox/acpt/ctem/bin/`)
- (`ctem-case;*.***` `/aaitp/fox/acpt/ctem/database/ifd/`)
- (`setf (lp::logical-pathname-translations "arpi")`
- (`("tachyon;*.***` `/aaitp/dana/arpi/tachyon/`)))

5.2 SIPE-2:IFD-4

Table 2 shows the files that constitute SIPE-2:ifd-4, and describes their contents. These files contain SIPE-2 extensions that are specific to IFD-4,^{*} and so are not part of the regular SIPE-2 distribution.

Table 2. Source Code Files for SIPE-2:ifd-4

FILE	DESCRIPTION
<code>init-ifd4.lisp</code>	Initializes the IFD-4 system, sets globals, etc.
<code>main.lisp</code>	Main functions and some global settings. Defines (1) the functions that are called by the user interface; (2) the top level functions for IFD-4; (3) the top level functions for running with advice, and (4) the top level functions for generating two plans and comparing them
<code>system.lisp</code>	Defines the IFD-4 system [*]
<code>translations.lisp</code>	Defines the logical pathname translations
<code>global.lisp</code>	Sets global variables to control system operation
<code>extract-arg.lisp</code>	Used to extract property values from SIPE-2 nodes
<code>status.lisp</code>	Accessors for node status indicators
<code>sipe-index.lisp</code>	Maps SIPE-2 class instances to numbers, for indexing
<code>target.lisp</code>	Translates the targets from a modified [†] <code>ctem_input</code> file into a SIPE-2 file with one object definition per target, and a SIPE-2 predicate file that contains one or more predicates per target
<code>compare.lisp</code>	Functions for criterion-based plan comparison
<code>backend.lisp</code>	Functions to connect to the user interface. Opens and tears down socket to the user interface, and parses user interface messages
<code>tiny.lisp</code>	Code to write the file for the text plan view

^{*}In Lisp, systems are used to organize related code and provide simplified loading, compiling, and version control.

[†]Modifications were needed to be made to the `ctem_input` file before using it in SIPE-2:ifd-4 were (1) removing header information, and (2) changing latitude/longitudes to numeric values, removing seconds and hemisphere.

The file `backend.lisp` contains a global variable that must be set after installation. The current setting is shown below, and should be changed to reflect the address or machine name of the machine on which the software will run.

```
(defparameter *motif-ip-address* "curie.erg.sri.com")
```

^{*}This is the software that was developed and delivered under Rome Laboratory Contract No. F30602-95-C-0175.

The code in `tiny.lisp` writes to the file `rawPlan.titles` as defined by the SIPE-2:ifd-4 Lisp variable `*TPV-PLAN-DUMP-FILE*`. Plan levels are indicated by integers, as follows: (1) Air Objective, (2) Task, (3) Activity, and (4) Support. A sample file format is shown below.

```
1Maintain Air Superiority in the AOR
2Disable-key-iads Libya
3Target al-jafra
4Support fighter_Sweep
```

There are no spaces after the numbers, and each objective is on one line. Hierarchical plan levels are indicated by increasing integer values. After this file is written into the `userinterface` directory, the executable command `CvtPlanFile` (also stored in the `userinterface` directory) is called so that this file may be reformatted into a form suitable for the Motif-based user interface. The result of `CvtPlanFile` is written in `PlanFile.txt`.

The file `pvt.lisp` writes the output files for display by PVT, and calls PVT to execute and display the generated plan. (The directory `tachyon-files` contains an Xwindows initialization file for PVT.) A global variable, `*PVT-IMAGE*`, is set in the file `global.lisp` and points to the executable Tachyon binary. Its current setting is shown below. This variable should not need to be changed as long as the PVT binary, called `xtach`, is in the correct directory.

```
(defvar *PVT-IMAGE* "arpi:tachyon;xtach")
```

The Tachyon binary is called by SIPE-2 during planning using the global variable `*TACHYON-3-IMAGE*`. This variable should not need to be changed as long as the binary for Tachyon is in the correct directory.

```
(setq *TACHYON-3-IMAGE* "arpi:tachyon;tach")
```

6 SIPE-2 AND ACPT

A separate directory, `acptinterface`, is provided for the functions needed to connect to ACPT and download a plan. The ACPT system, and the Node Object Manager code (written by ISX) in order to link ACPT to SIPE-2 is in the subdirectory called

```
acptinterface/JPT/JPTSystem/jpt/src/UI/cAgent.
```

This directory contains Lisp source code for the ACPT plan copy functions that are included in the SIPE-2:ifd-4 system definition. The files used for this integration are `lispcomm.lisp`, `clos-plan.lisp`, `make-plan.lisp`, `clos-agenda.lisp`, and `inspect-agenda-prefs.lisp`. Table 3 shows these files, written by SRI, that are in the `sipe-ifd4/lisp` directory for use in connecting SIPE-2:ifd-4 to ACPT.

Table 3. Source Code Files for SIPE-2:ifd-4

FILE	DESCRIPTION
<code>image-init.lisp</code>	Copies the ACPT plan and creates an image of it
<code>jpt2sipe.lisp</code>	Integrates the CLOS plan objects imported from ACPT in order to gain access to the objectives and activities that will be posted to SIPE-2 as goals
<code>tgt-tst.lisp</code>	Determines the size of the target set in the ACPT CLOS plan
<code>post-acpt.lisp</code>	Translates CLOS plan objects into SIPE-2 ACT representation

Instructions for downloading a plan from ACPT into SIPE-2:ifd-4 are provided in the file `acptinterface/README`. Because we do not distribute ACPT with SIPE-2:ifd-4, we cannot guarantee that the procedure described in that file is still functional, and we have no way to test it without cooperation from ISX. However, the procedure described in the `README` file worked with the IFD-4 final demonstration in February of 1997.

The distribution of SIPE-2:ifd-4 contains a saved Lisp image (in `sipe-ifd4/bin-sipe/ifd4-final`) that contains the CLOS plan that has already been downloaded from ACPT. This executable image should be used for testing and demonstration purposes. It contains SIPE-2, the ACPT plan (including any targets in the plan), and the code and knowledge base for IFD-4. If changes to the code or knowledge base are made, the IFD-4 system can be reloaded into this image by executing the Lisp command

```
(load "acp:lisp;init-ifd4").
```

7 SIPE-2 AND CTEM

SIPE-2 uses CTEM to provide additional information about strike packages and the timing of strikes in its air campaign plan. In the IFD-4 design, CTEM was to be called from SIPE-2:ifd-4 during run time. However, the execution time of CTEM took too long to reach completion. Therefore, SIPE-2:ifd-4 used files that were generated by a previous execution of CTEM.

There are two files containing code relevant to CTEM in `sipe-ifd4/lisp`. The first, `ctem.lisp`, contains code to connect Lisp to CTEM to retrieve its results, and the second, `ctem-sipe.lisp`, contains code to convert CTEM forms to SIPE-2 problems for planning.

The function in `ctem.lisp` scans a SIPE-2 plan to find activities that become part of the CTEM input before CTEM is run. They also convert CTEM output files so that support strike goals may be generated for SIPE-2:ifd-4 to solve.

The following list describes the interface to CTEM.

1. Generate a plan that has some "target" primitive actions in it.
2. Execute the Lisp function `(run-ctem)`.
3. Execute the Lisp function `(next-ctem-result)` repeatedly until `Nil` is returned.

The result of the execution of `next-ctem-result` is a list in the form `(NodeId. Goal)`. `NodeId` is the identifier of a node in the plan from which the target action is taken. `Goal` is a list that is isomorphic to a SIPE-2 goal that is to be constructed in order to support the strike on the given target. The list of CTEM results is collected in a variable called `*ctem-results*`. The functions in `ctem-sipe.lisp` use this list to create a SIPE-2 "planhead node," the newly created problem to be solved for these results. This procedure allows SIPE-2:ifd-4 to plan the required support missions for each CTEM result.

Several variations of the CTEM run for IFD-4 were created and saved for later use. These may be found in the directory `cteminterface/database/ifd`, and represent different resources and air tasks input into CTEM. These files are `117.ss` and `no117.ss`; they represent a situation in which F-117 aircraft were available, and unavailable, respectively.

When we initially designed IFD-4 such that CTEM would be called by SIPE-2 during the execution of the demonstration, we parsed the CTEM results in order to extract information required by SIPE-2:ifd-4 for planning. SIPE-2:ifd-4 also needed to tailor the input files for CTEM (using its required syntax) according to changes that had been made in the resources by means of the user interface. The directory `cteminterface/ctem-parser` contains the code for parsing the CTEM output files. CTEM places its output in files with the suffix ".mrv" and the parser translates them into files ending with ".ss". Input for CTEM execution is written by SIPE-2:ifd-4 into the file `cteminterface/database/ifd/ctem_input`. This file contains the targets that CTEM uses in its scheduling and packaging.

8 INSTALLING AND OPERATING THE IFD-4 SOFTWARE

The SIPE-2:ifd-4 software distribution is in the form of a tar file. When extracted, it creates the four directories is described above. Once extracted, the file `sipe-ifd4/lisp/translations.lisp` must be customized with mappings between logical paths and absolute paths linked to the locations in which the directory trees were installed, as described above. In addition to the contents of this tar distribution, version 4.13 of SIPE-2 must be installed, in Allegro Common Lisp 4.3.

To run the user interface binaries, the user must set the UNIX environment variable `LD_LIBRARY_PATH` to `/usr/openwin/lib:/usr/lib:/usr/dt/lib`. The user interface binaries are all dynamically linked. Source codes are also provided, but to link the table windows to the source code, a license is needed for the XRT/Table product, with which the table windows were created.

Running the IFD-4 and Plan Comparison Demonstration. The IFD-4 and plan comparison demonstration should be run in an X Windows environment on a Sun workstation running the Solaris operating system. In the Xwindow under which the SIPE-2 image will run, configure the PVT's user interface by loading its X Windows resources using the following command:

```
xrdb -merge /aaftp/dana/arpi/tachyon/XTach.Xdefaults.
```

Start the image in `sipe-ifd4/bin-sipe/sipe-ifd4`. This executable Lisp image contains the SIPE-2:ifd-4 code, the ACPT plan, and the SIPE-2:ifd-4 knowledge base. This will automatically load any SIPE-2 patches. Next, load the modified SIPE-2:ifd-4 logical pathnames by executing the following Lisp command in the Lisp listener window, replacing `[installation-path]` with the UNIX pathname indicating the location in which SIPE-2:ifd-4 was installed:

```
(load "[installation-path]/sipe-ifd4/lisp/translations.lisp").
```

Execute the following initialization Lisp command:

```
(in-package :sipe)
(load "acp:lisp;init-ifd4").
```

The second command will ensure that any changes to the source code will be loaded.

To run the demonstration, first bring up the user interface by executing

```
(demo-script-w-ui).
```

If this is the first time that this function is called, it will call SIPE-2 to create a short plan to initialize the links to the non-air superiority targets.

Once the control panel window has appeared, press the `Plan` button to instruct SIPE-2 to create the pre-CTEM plan. SIPE-2 will report progress in the Lisp listener window, and will restore the `Done` button in the control panel window to blue when it has completed this task. At this point, other features of the user interface could be used, as described above. Next, complete the plan by selecting the `Analyze` button. This will take 3–4 minutes with a complete set of targets (approximately 400). (It is normal to see numerous “No nodeid found” warnings in the Lisp listener window.) Upon completion of the plan, SIPE-2:ifd4 will invoke PVT, and the PVT windows will appear.

An example exploration of the Plan Visualization Tool would be from Objective to Support Mission. To begin this exploration, middle click on the gray bar of the first air objective for Weapons of Mass Destruction, as shown at the top of Figure 8. Then middle-click on the new, single task that is shown (see the middle of Figure 8). Middle-click on the two targets labeled “Al Jufra” (shown near the bottom of Figure 8) to see contrasting resource availabilities (these support tasks are shown in Figure 9). Before proceeding to the next step, exit the PVT.

To run the plan again, this time adding advice, return to the Lisp listener window and interrupt the top-level loop with `ctrl-C`. Then execute the following:

```
(use-advice).
```

The plan can now be rerun using the `Plan` and `Analyze` buttons, by continuing from the interrupt, as follows:

```
:cont 0.
```

(Lisp will report a warning about a select error that can be ignored.)

When planning is complete, again interrupt the loop with `ctrl-C` so that the comparison function may be run:

```
(display-all-comparisons).
```

This will bring up three windows, one showing aircraft assets common to both plans, the second showing those available only in one plan, and the third showing those available only in the other plan. These windows can be moved, scrolled, and resized manually, to view their contents.

To restore the system to its pre-demo state hit `ctrl-C` again to return to the Lisp prompt, bring down the user interface from Lisp, and restore the global defaults:

```
:pop  
(kill-ui)  
(setf *user-advice* nil).
```

The function `kill-ui` will remove the control panel window. You should close the comparison windows and any remaining Tachyon windows manually, to reduce clutter.

To show a demonstration similar to the original IFD-4, a demonstration in which plans with and without F-117s are compared, run the following from the Lisp prompt:

```
(setf *user-advice* nil)  
(117-demo).
```

Plan Objectives/Tasks		Status 0	20
AIR-OBJECTIVE: Destroy Weapons of Mass Destruction Ca			
AIR-OBJECTIVE: Destroy Petroleum Production of LIBYA			

Plan Objectives/Tasks		Status 0	20
AIR-TASK: Destroy WMD Delivery Vehicles of LIBY			

Plan Objectives/Tasks		Status 0	20
ACTIVITY: TARGET AL-KHUMS-9002WM0330-3			
ACTIVITY: TARGET AL-JUFRA-9002WM0326-80			
ACTIVITY: TARGET AL-JUFRA-9002WM0327-80			
ACTIVITY: TARGET GHURDABIYAH-9002WM033			
ACTIVITY: TARGET AL-RABDAH-9002WM0333-			

Figure 8. PVT Display Showing Three Levels of the Plan

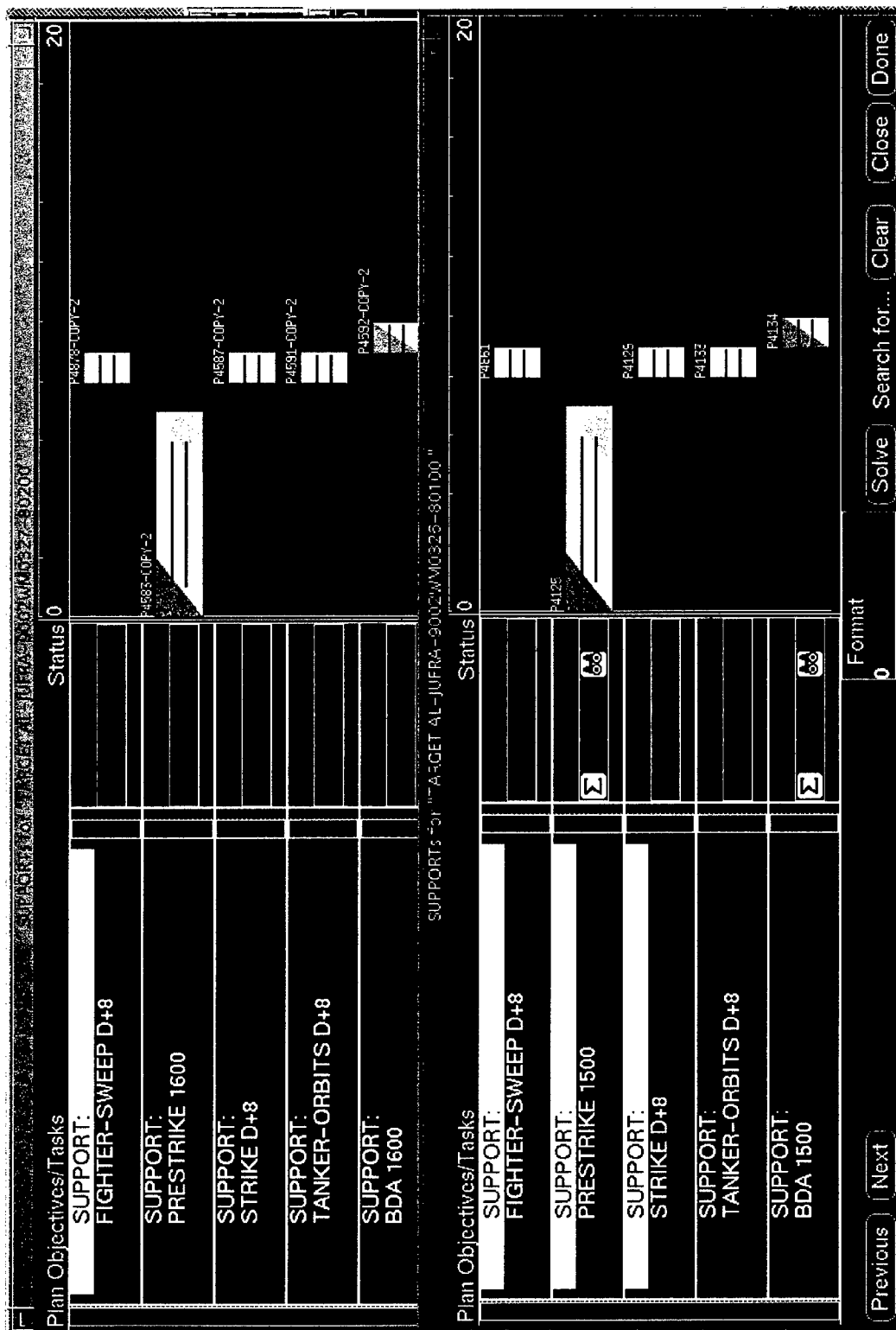


Figure 9. PVT Display Showing Support Missions Generated by SIPE-2:ifd4

This generates two full plans, and displays a comparison of the aircraft used by each. (It takes about 10 minutes.) The two plans are scheduled somewhat differently, so a detailed analysis of the comparison may reveal some anomalies. The comparison will show other aircraft taking over the activities of the F-117s. To test plan advice without using the user interface, the same two plans may be generated, the first without advice, and the second with fuel-economy advice; then run their comparison by executing

(advice-demo).

The comparison function (display-all-comparisons) may be run again afterwards to display the results.

9 REFERENCES

- Bienkowski, M.A. 1997. *SOCAP-ACPT Technology Integration and Application*, Technical Report ITAD-7112-AR-97-095, SRI International, Menlo Park, California (June).
- Bienkowski, M.A., T.J. Lee, and M.J. Wolverton. 1997. *Air Campaign Support Mission Knowledge Base* (in preparation), SRI International, Menlo Park, California.
- Myers, K. L. 1996. *User Guide for the Advisable Planner, Version 1.0*, Technical Report, SRI International Artificial Intelligence Center, Menlo Park, California (October).
- Wilkins, D.E. 1997. *Using the SIPE-2 Planning System: A Manual for SIPE-2, Version 4.14*, Revised Working Document (7 February).

DISTRIBUTION LIST

addresses	number of copies
WAYNE A. BOSCO AFRL/IFTB 525 BROOKS ROAD ROME NY 13441-4505	4
SRI INTERNATIONAL 333 RAVENSWOOD AVENUE MENLO PARK CA 94025-3493	5
AFRL/IFOIL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-DCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	2
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
DR JAMES ALLEN COMPUTER SCIENCE DEPT/BLDG RM 732 UNIV OF ROCHESTER WILSON BLVD ROCHESTER NY 14627	1
DR YIGAL ARENS USC-ISI 4676 ADMIRALTY WAY MARINA DEL RAY CA 90292	1
DR MARIE A. BIENKOWSKI SRI INTERNATIONAL 333 RAVENSWOOD AVE/EK 337 MENLO PRK CA 94025	1

DR MARK S. BOODY
HONEYWELL SYSTEMS & RSCH CENTER
3660 TECHNOLOGY DRIVE
MINNEAPOLIS MN 55418

1

DR MARK BURSTEIN
BBN SYSTEMS & TECHNOLOGIES
10 MOULTON STREET
CAMBRIDGE MA 02138

1

DR THOMAS L. DEAN
BROWN UNIVERSITY
DEPT OF COMPUTER SCIENCE
P.O. BOX 1910
PROVIDENCE RI 02912

1

DR PAUL R. COHEN
UNIV OF MASSACHUSETTS
COINS DEPT
LEDERLE GRC
AMHERST MA 01003

1

DR JON DOYLE
LABORATORY FOR COMPUTER SCIENCE
MASS INSTITUTE OF TECHNOLOGY
545 TECHNOLOGY SQUARE
CAMBRIDGE MA 02139

1

DR MICHAEL FEHLING
STANFORD UNIVERSITY
ENGINEERING ECO SYSTEMS
STANFORD CA 94305

1

RICK HAYES-ROTH
CIMFLEX-TEKKNOWLEDGE
1810 EMBARCADERO RD
PALO ALTO CA 94303

1

DR JIM HENDLER
UNIV OF MARYLAND
DEPT OF COMPUTER SCIENCE
COLLEGE PARK MD 20742

1

MR. MARK A. HOFFMAN
ISX CORPORATION
1165 NORTHCHASE PARKWAY
MARIETTA GA 30067

1

DR RON LARSEN 1
NAVAL CMD, CONTROL & OCEAN SUR CTR
RESEARCH, DEVELOP, TEST & EVAL DIV
CODE 444
SAN DIEGO CA 92152-5000

DR. ALAN MEYROWITZ 1
NAVAL RESEARCH LABORATORY/CODE 5510
4555 OVERLOOK AVE
WASH DC 20375

ALICE MULVEHILL 1
BBN
10 MOULTON STREET
CAMBRIDGE MA 02238

DR DREW MCDERMOTT 1
YALE COMPUTER SCIENCE DEPT
P.O. BOX 2158, YALE STATION
51 PROSPECT STREET
NEW HAVEN CT 06520

DR DOUGLAS SMITH 1
KESTREL INSTITUTE
3260 HILLVIEW AVE
PALO ALTO CA 94304

DR. AUSTIN TATE 1
AI APPLICATIONS INSTITUTE
UNIV OF EDINBURGH
80 SOUTH BRIDGE
EDINBURGH EH1 1HN - SCOTLAND

DIRECTOR 1
DARPA/ITO
3701 N. FAIRFAX DR., 7TH FL
ARLINGTON VA 22209-1714

DR STEPHEN F. SMITH 1
ROBOTICS INSTITUTE/CMU
SCHENLEY PRK
PITTSBURGH PA 15213

DR JONATHAN P. STILLMAN 1
GENERAL ELECTRIC CRD
1 RIVER RD, RM K1-5C31A
P. O. BOX 8
SCHENECTADY NY 12345

DR EDWARD C.T. WALKER
BBN SYSTEMS & TECHNOLOGIES
10 MOULTON STREET
CAMBRIDGE MA 02138

1

DR BILL SWARTOUT
USC/ISI
4676 ADMIRALTY WAY
MARINA DEL RAY CA 90292

1

DR MATTHEW L. GINSBERG
CIRL, 1269
UNIVERSITY OF OREGON
EUGENE OR 97403

1

MR JEFF GROSSMAN, CO
NCCOSC ROTE DIV 44
5370 SILVERGATE AVE, ROOM 1405
SAN DIEGO CA 92152-5146

1

DR ADELE E. HOWE
COMPUTER SCIENCE DEPT
COLORADO STATE UNIVERSITY
FORT COLLINS CO 80523

1

DR LESLIE PACK KAEHLING
COMPUTER SCIENCE DEPT
BROWN UNIVERSITY
PROVIDENCE RI 02912

1

DR SUBBARAO KAMBHAMPATI
DEPT OF COMPUTER SCIENCE
ARIZONA STATE UNIVERSITY
TEMPE AZ 85287-5406

1

DR MARTHA E POLLACK
DEPT OF COMPUTER SCIENCE
UNIVERSITY OF PITTSBURGH
PITTSBURGH PA 15260

1

DR MANUELA VELOSO
CARNEGIE MELLON UNIVERSITY
SCHOOL OF COMPUTER SCIENCE
PITTSBURGH PA 15213-3891

1

DR DAN WELD DEPT OF COMPUTER SCIENCE & ENG MAIL STOP FR-35 UNIVERSITY OF WASHINGTON SEATTLE WA 98195	1
DR TOM GARVEY DARPA/ISO 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
DIRECTOR ARPA/ISO 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
OFFICE OF THE CHIEF OF NAVAL RSCH ATTN: MR PAUL QUINN CODE 311 800 N. QUINCY STREET ARLINGTON VA 22217	1
DR GEORGE FERGUSON UNIVERSITY OF ROCHESTER COMPUTER STUDIES BLDG, RM 732 WILSON BLVD ROCHESTER NY 14627	1
DR STEVE HANKS DEPT OF COMPUTER SCIENCE & ENG'G UNIVERSITY OF WASHINGTON SEATTLE WA 98195	1
DR ADNAN DARWICHE INFORMATION & DECISION SCIENCES ROCKWELL INT'L SCIENCE CENTER 1049 CAMINO DOS RIOS THOUSAND OAKS CA 91360	1
MR. ROBERT J. KRUCHTEN HQ AMC/SCA 203 W LOSEY ST, SUITE 1016 SCOTT AFB IL 62225-5223	1
DR. MAREK RUSINKIEWICZ MICROELECTRONICS & COMPUTER TECH 3500 WEST BALCONES CENTER DRIVE AUSTIN, TX 78759-6509	1

MAJOR DOUGLAS DYER/ISO
DEFENSE ADVANCED PROJECT AGENCY
3701 NORTH FAIRFAX DRIVE
ARLINGTON, VA 22203-1714

1

DR. STEVE LITTLE
MAYA DESIGN GROUP
2100 WHARTON STREET S&E 702
PITTSBURGH, PA 15203-1944

1

NEAL GLASSMAN
AFDSR
110 DUNCAN AVENUE
BOLLING AFB, WASHINGTON, D.C.
29332

1